

## Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application.

### Listing of Claims:

#### Claim 1 (currently amended)

1. A computer-implemented ~~based~~ method for graphical development of fully executable workflow applications, wherein said fully executable workflow applications are stored in a form of computer-executable program code ~~for loading in computer memory code segments and, after said computer-executable program code being loaded and, after said workflow applications being run, a processor runs the workflow operation~~ by invoking said program code ~~loaded in code segments~~, said program code facilitates processing of workflow orders, wherein said method for graphical development comprises the following steps:

(A) Describing an element of workflow-process graphically with software that accepts interactive user input;

(B) Interpreting the ~~accepted at step (A)~~ user input to obtain workflow-related data and transforming obtained workflow-related data into an incremental modification of incrementally constructed set of class objects that represents a workflow-process description;

(C) Repeating ~~step~~ steps (A) and ~~step~~ (B) in a loop until workflow-process description is complete;

(D) Extracting workflow-process-related data from the ~~produced at step (B)~~ set of class objects and transforming the extracted workflow-process-related data into a workflow-process definition;

(E) Generating source code of the workflow application software, and compiling instructions, and linking instruction from the produced-at-step (D) workflow-process definition;

(F) Compiling the generated-at-step (E) source code of the workflow application software with the generated-at-step (E) compiling instructions;

(G) Linking result of the performed at-step (F) compiling with generated at-step (E) linking instructions;

and

wherein said method further comprises defining a workflow-process comprising the following steps:

(a) Establishing a matrix of workflow-activities, wherein said workflow-activities are individual items of work comprised by a potential workflow-process where all transactional operations potentially performed by each one of said individual items are potentially executed as a single transaction within each one of said workflow-activities of said potential workflow-process, and wherein said establishing the matrix further comprises:

(i) Arranging workflow-activities comprised by the workflow-process being defined according to required sequence of execution of said workflow-activities within said workflow-process and according to possibilities for concurrent execution of some of said workflow-activities within said workflow-process;

(ii) Defining a constant, named PROCESS\_STEPS, representing a number of sequential steps of execution of a workflow-process being defined, wherein on each one of said sequential steps of execution either one activity is executed, or either two or more activities are executed concurrently;

(iii) Defining a constant, named MAX\_STEP\_DEPTH, representing a maximum number of workflow-activities executed concurrently on one step of workflow-process being defined;

(iv) Declaring and initializing a two-dimensional array of Boolean elements being a workflow-activities matrix, with dimensions MAX\_STEP\_DEPTH by PROCESS\_STEPS, wherein an array element with value TRUE represents existence of a workflow-activity on position in said workflow-activities-matrix with coordinates equal to said array element indexes and wherein an array element with value FALSE represents non-existence of a workflow-activity on position in said workflow-activities with coordinates equal to said array element indexes;

(b) Defining a main flow-graph, wherein the defining matrix of workflow-activities are further being represented as nodes of said flow-graph by incorporating control-connectors in order to represent a potential flow-of-control between said nodes within a potential workflow-process, wherein each said control-connector signals successful execution of its sourcing workflow-activity with execution result having value TRUE, wherein two or more control-connectors sourcing from one of said nodes initiate concurrent execution of destination nodes of said control-connectors, wherein two or more control-connectors having a common destination node impose a requirement for synchronizing conjunction of signals of all control-connectors incoming to said common destination node in order to trigger execution of said common destination node, wherein said defining the main flow-graph further comprises:

(i) Defining a constant, named MAX\_CONNECT\_OUT, representing maximum number of main control-connectors sourcing from one flow-graph node representing a workflow-activity in workflow-process being defined;

(ii) Declaring and initializing a two-dimensional array of integer elements with dimensions MAX\_CONNECT\_OUT by 2 per workflow-activity for each one of activities belonging to workflow steps with numbers from 1 to (PROCESS\_STEPS-1), wherein values of each pair of elements of said array represent indexes of the initialized two-dimensional array of workflow-activities and thereby describe a position of a workflow-activity in said workflow-activities matrix,

wherein said described positions of MAX\_CONNECT\_OUT workflow-activities in said workflow-activities matrix are positions of destination workflow-activities of MAX\_CONNECT\_OUT control-connectors having common source workflow-activity with potential execution result having value TRUE;

(c) Defining one or more alternative control connectors, wherein each one of said alternative control connectors signals successful execution of its sourcing workflow-activity with execution result having value FALSE, wherein two or more of said alternative control-connectors sourcing from same workflow-activity initiate concurrent execution of destination workflow-activities of said alternative control-connectors, wherein a workflow-activity having established during the defining main flow-graph a requirement for synchronizing conjunction of signals of all incoming to it control-connectors cannot be a destination workflow-activity of an alternative control connector, wherein defining the one or more alternative control-connectors further comprises:

(i) Defining a constant, named MAX\_ALTCONNECT\_OUT, representing maximum number of alternative control-connectors sourcing from one flow-graph node representing a workflow-activity in workflow-process being defined;

(ii) Declaring and initializing a two-dimensional array of integer elements with dimensions MAX\_ALTCONNECT\_OUT by 2 per workflow-activity for each one of activities belonging to workflow steps with numbers from 1 to (PROCESS\_STEPS-1), wherein values of each pair of elements of said array represent indexes of the initialized two-dimensional array of workflow-activities and thereby describe the position of a workflow-activity in said workflow-activities matrix, wherein said described positions of MAX\_ALTCONNECT\_OUT workflow-activities in said workflow-activities matrix are positions of destination workflow-activities of MAX\_ALTCONNECT\_OUT control-connectors having common source workflow-activity with potential execution result having value FALSE;

(d) Defining workflow-components matrix, wherein every element of said workflow-components matrix represents a software component, associated

with a workflow-activity of workflow-process being defined, for potential plugging for execution as part of a potential single transaction within said workflow-activity of the workflow-process, wherein the defining of the workflow-component matrix further comprises:

(i) Declaring a two-dimensional array of elements of type UUID (Universal Unique Identifier, interchangeable with the term GUID, Globally Unique Identifier) with dimensions identical to dimensions of the declared two-dimensional array representing workflow-activities matrix;

(ii) Initializing elements of the declared array in the following manner: for every element of said array representing the workflow-activities matrix with value TRUE to be initialized corresponding element with identical indexes in array representing workflow-components matrix with UUID value representing a software component to be associated with represented workflow-activity, and for every element of said array representing workflow-activities matrix with value FALSE to be initialized corresponding element with identical indexes in array representing workflow-components matrix with UUID value indicating a dummy UUID.

Claim 2 (cancelled)

Claim 3 (currently amended)

3. A computer-implemented method for graphical development of fully executable workflow applications as per claim 2 1, ~~further including method for generation of source code and compiling and linking instructions, wherein said source code and compiling and linking instructions are necessary and sufficient to build fully executable workflow application with third party produced and available on the market compilers and linkers, wherein~~ said method for generation further involves interpreting workflow process definition data and a parameterized executing of functions that provide software source code building blocks, wherein said parameterized executing of functions comprises:

- (A) Executing of functions that build software source code skeleton;
- (B) Executing of a function that builds part of software source code that declares a set of variables and initializes these variables with relevant data from said workflow-process definition;
- (C) Executing of a function that builds part of software source code verifying that the necessary workflow components according to the workflow process definition data are registered and available on a computer where workflow application will potentially be executed;
- (D) Executing of a function that builds part of the software source code responsible for the construction of class objects that represent workflow configuration according to the workflow-process definition data;
- (E) Executing of a function that builds compiling and linking instructions.

Claim 4 (cancelled)

Claim 5 (currently amended)

~~5. A system for transactional processing of workflow, wherein skeleton of said system is a hierarchical tree of class objects with capacity to represent variety of workflow configurations, wherein said class objects are computer memory instances of classes or structures, wherein any level of said hierarchical tree contains one or multiple collections, each said collection containing predefined or dynamically defined number of one or multiple objects, wherein objects belonging to any said collection of same hierarchy level are computer memory instances of classes or structures of identical type, wherein first level of said hierarchical tree consists of a single collection and every next level of hierarchy contains one or multiple collections and possibly individual objects, wherein said collections and said individual objects being accessible via references controlled by objects belonging to collections from previous level, wherein said hierarchical tree of class objects contains following levels~~

and siblings: A software product stored in a computer storage medium, executed by processor for transactional processing of workflow comprising:

(A) System skeleton as a hierarchical tree of class objects with capacity to represent a variety of workflow configurations, wherein:

- Said class objects are computer memory instances of classes or structures;

- Any level of said hierarchical tree contains one or multiple collections and each said collection contains a predefined or dynamically defined number of one or multiple objects, wherein objects belonging to any said collection of same hierarchy level are computer memory instances of classes or structures of identical type;

- First level of said hierarchical tree consists of a single collection and every next level of hierarchy contains one or multiple collections and individual objects, wherein said collections and said individual objects are accessible via references controlled by objects belonging to collections from previous level;

Wherein said hierarchical tree of class objects contains the following levels and siblings:

(i) First level, a steps-collection, wherein collection objects represent steps of said workflow-process, each said object being parent of a Second level collection;

(ii) Second level, an activities-collection per step of said workflow-process, wherein said collection objects represent workflow-activities that might be executed concurrently with all other workflow-activities represented by objects belonging to the same collection, wherein each object is a parent of Third level siblings, said siblings being of 3 distinct categories: a processor collection, a workload-dispatching balancer object, and where necessary a workflow-synchronization object;

(iii) Third level's sibling One, a processor-collection per workflow-activity, wherein an object of said collection represents an individual workload-processing squad having a capacity to process a portion of entire workload specific for said workflow-activity, each said object being parent of a Fourth level collection;

(iv) Third level's sibling Two, a workload-dispatching object per workflow-activity;

(v) Third level's ~~optional~~ sibling Three, a workflow-synchronizing object per workflow-activity if required by flow-graph of said workflow-process, wherein said object contains data defining required synchronization scheme;

(vi) Fourth level, a processing-thread-dedicated objects collection, where an object of said collection represents a dedicated area for data holding and data exchange between threads.

(B) Means for splitting of a control flow by launching one or more new parallel control flows, wherein said parallel control flow comprises execution of at least one workflow-activity, or sequence of plurality of workflow-activities, before joining with its launching, or other, flow of control, wherein said parallel control flow launches one or more new parallel control flows, wherein a workflow-activity, that is part of a parallel control flow, generates notification message to synchronizing thread at control flow join-point with other control flow or plurality of control flows about a potential result with value FALSE of its execution, wherein a workflow-activity that is part of a parallel control flow further splitting one or more times, generates notification messages to synchronizing threads at splitting branches join-points with other control flows about a potential result with value FALSE of its execution, wherein synchronizing threads generate notification messages to synchronizing threads responsible for join-point of synchronized control flow with one or more control flows or to synchronizing threads responsible for join-points of said synchronized control flow's splitting branches about termination of said synchronized control flow.



Claim 6 (cancelled)

Claim 7 (currently amended)

7. A ~~system for transactional processing of workflow~~ software product as per claim 6 5, further including apparatus means for synchronization of two or more parallel control flows before execution of next in flow-graph workflow-activity according to a synchronization scheme, wherein said synchronization scheme is a conjunction of events signaling completed execution of all workflow-activities represented by sourcing nodes of two or more flow-graph control connectors with destination node being said workflow-activity that will be executed after said synchronization as part of a potential single workflow-request, wherein implementation of said means for synchronization applies to entire multitude of concurrently processed potential workflow-requests.

Claim 8 (currently amended)

8. A ~~system for transactional processing of workflow~~ software product as per claim 7, further including apparatus means for launching of alternative control flow routes, wherein said launching takes place where potential execution of a workflow-activity has a result with value FALSE, wherein said alternative control flow routes redirect workflow execution from its normal routes by transferring execution control to one or more workflow-activities with non-synchronized execution, wherein an alternative control-connector launching an alternative control flow route and bypassing one or more workflow-activities with synchronized execution, is coupled with notification-connectors having same source node as said alternative control-connector, wherein each one of said notification-connectors has one of bypassed workflow activities with synchronized execution as its destination node.

Claim 9 (currently amended)

9. A ~~system for transactional processing of workflow~~ software product as per claim 8, further including construction of hierarchical structure of threads with four levels providing capacity for concurrent processing of multitude of workflow instances, wherein in said hierarchical structure levels below top level are organized as multitude of horizontally arranged divisions, wherein each said division is

autonomous and self-contained in conducting its tasks, wherein top level thread is responsible for making adaptive decisions, and executing and supervising adaptive behavior related to allocation and de-allocation of computer and operating system resources based on its own assessment of application needs, wherein said capacity for concurrent processing of multitude of workflow instances is limited only by environmental factors such as availability of reserve of system memory and unused CPU power and ability of networking infrastructure to cope with generated traffic, wherein said hierarchical structure of threads contains following levels:

- (A) First level, formed by created and activated processing threads, wherein threads forming this level of said hierarchical structure are directly responsible for transactional processing of requests for work and for transactional flow of control between workflow-activities;
- (B) Second level, formed by created and activated supervising threads, wherein threads forming this level of said hierarchical structure are responsible for assignment of requests to individual processing threads and supervision of requests' execution;
- (C) Third level, formed by all dispatching or synchronizing-dispatching threads being created and activated according to number of workload-dispatching objects in hierarchical tree of class objects and number of ~~optional~~ workflow-synchronizing objects in hierarchical tree of class objects, wherein dispatching threads being part of Third level are responsible for dispatching of requests to supervising threads, wherein synchronizing-dispatching threads being part of Third level are responsible for synchronization of execution of parallel control flows and for dispatching of requests to supervising threads;
- (D) Fourth level, wherein Fourth level is hierarchy's top level and comprises only one thread being responsible for making, executing and supervising decisions about allocation and de-allocation of system resources based on its own assessment, wherein said allocation and de-allocation takes form of modification of First and Second levels of hierarchical structure of threads and their corresponding objects and collections of hierarchical tree of class objects.

Claim 10 (currently amended)

10. A computer-implemented method for transactional plugging of software components ~~component~~ into a workflow-activity of a workflow-process at runtime, wherein said transactional plugging facilitates inclusion of all transactional operations, ~~potentially~~ performed by invoking a method of plugged software component, into each ~~potential~~ transaction initiated by each ~~potential~~ workflow instance for execution of said workflow-activity, wherein said method for transactional plugging comprises following the steps of:

(A) Creation and initialization of processing threads, wherein during its initialization each said processing thread instantiates a non-transactional component object and sends to it an amount of data being constant between processing of individual workflow requests and being necessary to perform execution of relevant portion of workflow request that will potentially be assigned to said processing thread;

(B) The instantiated ~~at step (A)~~ non-transactional component object instantiates its own transactional component-intercepting object, and with GUID (Globally Unique Identifier) of a workflow-activity software component, sent to it as a parameter, instantiates a workflow-activity component-intercepting object;

(C) Supervising thread receives potentially arriving multitudes of single workflow requests and assigns each one of said requests to an individual processing thread of its pool;

(D) A Processing thread having an assigned ~~at step (C)~~ request for work invokes a method of its non-transactional component object;

(E) ~~An invoked at step (D) non-transactional component object method calls a method of its~~ Calling a method of the transactional component-intercepting object that belongs to the non-transactional component object, wherein said call of transactional component-intercepting object method constructs a transactional component object thereby creating a new transaction and calls a method of said transactional component;

~~(F) Within context of the created at step (E) new transaction, said transactional component object method calls~~ Calling ~~a method of the instantiated at step (B) workflow-activity component-intercepting object by the transactional component object, wherein said method call instantiates a workflow-activity software component and invokes a~~ its ~~method of said component, which performs all of its transactional operations as members of the created new transaction.~~

**Claim 11 (currently amended)**

**11. A ~~system for transactional processing of concurrent workflow~~ software product as per claim 9, further including ~~apparatus~~ means for workload balancing structured at two levels, wherein upper level of said workload balancing comprises multitude of associations between a dispatching thread and multitude of supervising threads and involves dispatching thread balancing workload between its associated supervising threads, wherein lower level of said workload balancing comprises multiple groupings of processing threads in pools associated with a supervising thread per pool and involves supervising threads balancing workload between processing threads of their associated pools.**

**Claim 12 (currently amended)**

**12. A ~~system for transactional processing of concurrent workflow~~ software product as per claim 11, further including ~~apparatus~~ means for software bottlenecks' prevention and neutralizing, wherein said software bottlenecks' prevention involves encapsulation of a thread pool containing fixed number of processing threads with a supervising thread in a processing-pipe, wherein said software bottlenecks neutralizing comprises construction of additional processing-pipes and inclusion of constructed additional processing-pipes in workload balancing process related to workflow-activity where development of bottleneck has been detected.**

**Claim 13 (currently amended)**

**13. A ~~system for transactional processing of concurrent workflow~~ software product as per claim 12, further including ~~apparatus~~ means for detection of conditions requiring scaling up of available capacity for processing of concurrent workflow, wherein said detection in regard to a particular workflow-activity involves checking**

for conjunction of events, from all processing-pipes associated to said workflow-activity, signaling that number of idle threads in processing-pipe's pool reached its critical minimum.

**Claim 14 (currently amended)**

14. A ~~system for transactional processing of concurrent workflow~~ software product as per claim 13, further including ~~apparatus~~ means for scaling up of available capacity for processing of concurrent workflow, wherein said scaling up is triggered at a particular workflow-activity to counteract development of a bottleneck at that particular workflow-activity and triggered at all workflow-activities for higher workflow processing responsiveness when workload increases, wherein said scaling up involves creation and activation of an additional processing-pipe and inclusion of said additional processing-pipe in workload balancing scheme.

**Claim 15 (currently amended)**

15. A ~~system for transactional processing of concurrent workflow~~ software product as per claim 14, further including ~~apparatus~~ means for detection of conditions requiring scaling down of available capacity for processing of concurrent workflow, wherein said detection in regard to a particular workflow-activity involves checking for conjunction of events, from all processing-pipes associated to said workflow-activity, signaling that number of busy threads in processing-pipe's pool reached its critical minimum.

**Claim 16 (currently amended)**

16. A ~~system for transactional processing of concurrent workflow~~ software product as per claim 15, further including ~~apparatus~~ means for scaling down of available capacity for processing of concurrent workflow, wherein said scaling down is triggered to counteract a detected inefficiency in use of system memory and CPU time-slice allocated to threads.

**Claim 17 (currently amended)**

17. A ~~system~~ software product as per claim 16, further including class objects for real-time visualization of quantity, structure, and utilization of threads forming First

and Second levels of hierarchical structure of threads and said hierarchical structure's adaptation-enacted modifications of its First and Second levels, wherein said visualization serves as indicator of workload volume, indicator of points of delay caused by distributed infrastructure, and for observation and analysis of adaptive behavior of hierarchical structure of threads.

Claim 18 (currently amended)

~~18. A computer program product readable storage medium storing a set of instructions capable of being executed by a processor, wherein during its execution by a processor said set of instructions accepts interactive user input for graphical development description of fully executable workflow applications process as per claim 1, creates workflow process definition as per claim 2, and generates source code and compiling and linking instructions sufficient to build fully executable workflow applications as per claim 3.~~ A computer storage medium containing instructions, said instructions when executed by a processor carry out the methods of:

Accepting interactive user input for a graphical description of a workflow process, the instructions further performing:

(A) Describing an element of workflow-process graphically with software that accepts interactive user input;

(B) Interpreting the user input to obtain workflow-related data and transforming obtained data into an incremental modification of incrementally constructed set of class objects that represents workflow-process description;

(C) Repeating steps (A) and (B) in a loop until workflow-process description is complete.

(D) Extracting workflow-process-related data from the set of class objects and transforming the extracted workflow-process-related data into a workflow-process definition;

(E) Generating source code of the workflow application software, compiling instructions, and linking instruction from the workflow-process definition;

(F) Compiling the source code of the workflow application software with the compiling instructions;

(G) Linking result of the performed compiling with generated linking instructions.

Creating workflow process definition comprising:

(a) Establishing a matrix of workflow-activities, wherein said workflow-activities are individual items of work comprised by a potential workflow-process where all transactional operations potentially performed by each one of said individual items are potentially executed as a single transaction within each one of said workflow-activities of said potential workflow-process, and wherein said establishing the matrix further comprises:

(i) Arranging workflow-activities comprised by the workflow-process being defined according to required sequence of execution of said workflow-activities within said workflow-process and according to possibilities for concurrent execution of some of said workflow-activities within said workflow-process;

(ii) Defining a constant, named PROCESS\_STEPS, representing a number of sequential steps of execution of a workflow-process being defined, wherein on each one of said sequential steps of execution either one activity is executed, or either two or more activities are executed concurrently;

(iii) Defining a constant, named MAX\_STEP\_DEPTH, representing a maximum number of workflow-activities executed concurrently on one step of workflow-process being defined;

(iv) Declaring and initializing a two-dimensional array of Boolean elements being a workflow-activities matrix, with dimensions MAX\_STEP\_DEPTH by PROCESS\_STEPS, wherein an array element with value TRUE represents existence of a workflow-activity on position in said workflow-activities-matrix with coordinates equal to said array element indexes and wherein an array element with value FALSE represents non-existence of a workflow-activity on position in said workflow-activities with coordinates equal to said array element indexes.

(b) Defining a main flow-graph, wherein the defining matrix of workflow-activities are further being represented as nodes of said flow-graph by incorporating control-connectors in order to represent a potential flow-of-control between said nodes within a potential workflow-process, wherein each said control-connector signals successful execution of its sourcing workflow-activity with execution result having value TRUE, wherein two or more control-connectors sourcing from one of said nodes initiate concurrent execution of destination nodes of said control-connectors, wherein two or more control-connectors having a common destination node impose a requirement for synchronizing conjunction of signals of all control-connectors incoming to said common destination node in order to trigger execution of said common destination node, wherein said defining the main flow-graph further comprises:

(i) Defining a constant, named MAX\_CONNECT\_OUT, representing maximum number of main control-connectors sourcing from one flow-graph node representing a workflow-activity in workflow-process being defined;

(ii) Declaring and initializing a two-dimensional array of integer elements with dimensions MAX\_CONNECT\_OUT by 2 per workflow-activity for each one of activities belonging to workflow steps with numbers from 1 to (PROCESS\_STEPS-1), wherein values of each pair of elements of said array represent indexes of the initialized two-dimensional array of workflow-activities and - thereby describe a position of a workflow-activity in said



workflow-activities matrix, wherein said described positions of MAX\_CONNECT\_OUT workflow-activities in said workflow-activities matrix are positions of destination workflow-activities of MAX\_CONNECT\_OUT control-connectors having common source workflow-activity with potential execution result having value TRUE;

(c) Defining one or more alternative control connectors, wherein each one of said alternative control connectors signals successful execution of its sourcing workflow-activity with execution result having value FALSE, wherein two or more of said alternative control-connectors sourcing from same workflow-activity initiate concurrent execution of destination workflow-activities of said alternative control-connectors, wherein a workflow-activity having established during the defining main flow-graph a requirement for synchronizing conjunction of signals of all incoming to it control-connectors cannot be a destination workflow-activity of an alternative control connector, wherein defining the one or more alternative control-connectors further comprises:

(i) Defining a constant, named MAX\_ALTCONNECT\_OUT, representing maximum number of alternative control-connectors sourcing from one flow-graph node representing a workflow-activity in workflow-process being defined;

(ii) Declaring and initializing a two-dimensional array of integer elements with dimensions MAX\_ALTCONNECT\_OUT by 2 per workflow-activity for each one of activities belonging to workflow steps with numbers from 1 to (PROCESS\_STEPS-1), wherein values of each pair of elements of said array represent indexes of the initialized two-dimensional array of workflow-activities and – thereby describe the position of a workflow-activity in said workflow-activities matrix, wherein said described positions of MAX\_ALTCONNECT\_OUT workflow-activities in said workflow-activities matrix are positions of destination workflow-activities of MAX\_ALTCONNECT\_OUT control-connectors having common source workflow-activity with potential execution result having value FALSE;

(d) Defining workflow-components matrix, wherein every element of said workflow-components matrix represents a software component, associated with a workflow-activity of workflow-process being defined, for potential plugging for execution as part of a potential single transaction within said workflow-activity of the workflow-process, wherein the defining of the workflow-component matrix further comprises:

(i) Declaring a two-dimensional array of elements of type UUID (Universal Unique Identifier, interchangeable with the term GUID, Globally Unique Identifier) with dimensions identical to dimensions of the declared two-dimensional array representing workflow-activities matrix;

(ii) Initializing elements of the declared array in the following manner: for every element of said array representing the workflow-activities matrix with value TRUE to be initialized corresponding element with identical indexes in array representing workflow-components matrix with UUID value representing a software component to be associated with represented workflow-activity, and for every element of said array representing workflow-activities matrix with value FALSE to be initialized corresponding element with identical indexes in array representing workflow-components matrix with UUID value indicating a dummy UUID;

and

Generating source code and compiling and linking instructions sufficient to build fully executable workflow applications by:

(A) Executing of functions that build software source code skeleton;

(B) Executing of a function that builds part of software source code that declares a set of variables and initializes these variables with relevant data from said workflow-process definition;

(C) Executing of a function that builds part of software source code verifying that the necessary workflow components according to workflow-process definition data are registered and available on a computer where the workflow application will potentially be executed;

(D) Executing of a function that builds part of the software source code responsible for the construction of class objects that represent workflow configuration according to the workflow-process definition data;

(E) Executing of a function that builds compiling and linking instructions.

Claim 19 (cancelled)

Claim 20 (cancelled)